

10/parts

10/533161  
JC20 Rec'd PCT/PTO 29 APR 2005

## TITLE

Apparatus, Method, and Article of Manufacture for Visualizing Patterns of Change and Behavior On A Compute Infrastructure

## INVENTORS

David Nocera, Lorelei Wagner

## CROSS REFERENCE TO RELATED APPLICATION(S)/CLAIM OF PRIORITY

This application claims the benefit of US Application Number 60/422,005, filed October 29, 2002, which is incorporated in its entirety herein.

This application also relates and incorporates by reference in its entirety International Application Number PCT/US 02/18473, entitled "Apparatus, Method, and Article of Manufacture for Managing Change on a Compute Infrastructure," filed June 11, 2002.

## STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not applicable.

## REFERENCE OF AN APPENDIX

Not applicable.

## FIELD OF THE INVENTION

1        The present invention relates generally to compute and/or network management and more  
2        particularly to an improved system, method, apparatus, and article of manufacture for visualizing  
3        patterns of changes and behavior on a compute infrastructure such as the one shown in Figure 10.

4

5        **BACKGROUND OF THE INVENTION**

6        Heretofore, compute infrastructure change visualization techniques involve programmed  
7        alerting generated by user defined events on individual technology components or processes.

8        Determining what components have changed and isolating patterns of failure has been the  
9        responsibility of the individuals tasked with responding to alarms. As expected, the process is  
10      often time-consuming and cumbersome.

11      Furthermore, the existing focus of alerts on component or process failures undermines the  
12      ability of individuals to identify components with a pattern of success.

13      Accordingly, what is needed is a comprehensive way to visualize change on a compute  
14      infrastructure, and more particularly, a solution that detects and presents patterns of both positive  
15      and negative change on a compute infrastructure.

16

17      **SUMMARY OF THE INVENTION**

18      The present invention (also called Differential View) addresses the aforementioned  
19      problems of the prior art by providing for, among other things, an improved apparatus, method  
20      and article of manufacture for visualizing patterns of change and behavior on a compute  
21      infrastructure. Differential View provides for complete visualization of infrastructure change

1 and behavior and further provides interactive filters that identify and display patterns of change  
2 and behavior, on a graduated scale, for the compute infrastructure as a whole and for specific  
3 groups within the infrastructure.<sup>1</sup>

4 Other aspects, features and advantages of the present invention will become better  
5 understood with regard to the following description and accompanying drawings.

6

## 7 BRIEF DESCRIPTION OF THE DRAWINGS

8 Referring briefly to the drawings, exemplary embodiments of the present invention will  
9 be described with reference to the accompanying drawings in which Figures 1-10 graphically  
10 illustrate certain aspects and features of the present invention.

11

## 12 DETAILED DESCRIPTION OF THE INVENTION

13 Referring more specifically to the drawings, for illustrative purposes aspects of the  
14 present invention is depicted in the exemplary embodiments generally shown in Figures 1 - 10. It  
15 will be appreciated that the illustrated embodiments may vary as to their details, for example,  
16 representative icons (a square may be a circle), configuration (the exact screen layout may be  
17 adjusted), etc., without departing from the basic concepts disclosed herein. The following  
18 description, therefore, should not to be taken in a limiting sense.

19

## 20 High Level Description

---

<sup>1</sup> This allows any type of compute data to be consolidated and visualized; this view can occur pre- or post- database load, or without ever loading data to a database. Furthermore, the attribute-values may represent any defined test

1       Figure 1 illustrates a graphical representation of an exemplary embodiment of the present  
2 invention. As shown, the graphical view includes several underlying support mechanisms  
3 including: Colorized Grid of Nodes<sup>2</sup> (Fig 1 - 1.0): a map of nodes being monitored, grouped for  
4 ease of association (in this example, the white lines in the grid divide the nodes by location)  
5 colored by evaluation of change status; Baselines (Fig 1 - 2.0): a selection of sets of predefined  
6 node attribute values with which to evaluate node conformity; Groups (Fig 1 - 3.0): user defined  
7 node groupings for change and behavior pattern isolation; Pie Charts (Fig 1 – 4.0, 4.1): for  
8 providing quantitative percentage of change within the selected set of nodes for referential  
9 comparison; Time Frame (Fig 1 – 5.0, 5.1, 5.2): utilities from which to alter the time frame  
10 evaluated and presented; Auto Focus (Fig 1 – 6.0): a utility which evaluates the groups to present  
11 those with the greatest deviation from expected values; Custom Color (Fig 1 – 7.0): a utility to  
12 select the colors in which the graduated values for change appear; Rotate (Fig 1 – 8.0): providing  
13 view control; Create Report (Fig 1 – 9.0): a report generator.

14

15      Visualization

16       Figure 2 illustrates the group selection progression of functionality listed in the  
17 description of Figure 1. It presents the group pattern identification process which consists of the  
18 primary graphical view and supporting mechanisms: Selection of Groups (Fig 2 – 1.0), select the  
19 group to be distinguished from the enterprise node view; Identification of Nodes within Group  
20 Selection (Fig 2 – 2.0), nodes which belong to the selected Node Group are highlighted to be

---

(unit, system, performance, or industrial process).

1 distinguished from the full population of nodes; Group Selection Pie Chart (Fig 2 – 3.0) provides  
2 visualization of the quantitative percentage of change within the selected set of nodes; Node  
3 View Pie Chart (Fig2 – 4.0) provides visualization of the quantitative percentage of change in  
4 full population to provide a basis with which to compare the group to the whole. This ability  
5 provides a means by which to isolate the groups with the highest rate of change. The Auto Focus  
6 button (Fig 2 – 5.0) when clicked, will automatically select and present the group with the most  
7 significant rate of change.

8 Figure 3 progresses beyond group selection and into analysis of the group selection  
9 through Baseline Comparison.<sup>3</sup> Selection of Groups (Fig 3 – 1.0), select the group to be  
10 distinguished from the enterprise node view; Selection of Baseline (Fig 3 – 2.0), select the  
11 Baseline through which to filter the node group (this example provides a visualization of nodes  
12 in WEB-GRP1 and how they align with the pre-established attribute-value pairs in the WEB-  
13 PATCHES Baseline). Node View (Fig 3 – 3.0) presents the group nodes with the status relative  
14 to the Baseline; Node View Pie Chart (Fig 3 – 4.0) continually provides visualization of the  
15 quantitative percentage of change in full population. Group Selection Pie Chart (Fig 3 – 5.0)  
16 provides visualization of the quantitative percentage of change within the baseline for the  
17 selected set of nodes (in this example, 100% of WEB-GRP1 exactly match the WEB-PATCHES

---

<sup>2</sup> The concept of Node is not limited to a physical object and can be extended to a logical concept like a business process, object or application.

<sup>3</sup> It is not necessary to select a Group in order to select a baseline. One could look at a Baseline for patterns of change or behavior across the enterprise node view; however, patterns are more easily tracked when using both the Baseline and a Group. Figure 3 and 4 combined illustrate the use of Baseline compare to quickly analyze and isolate the set of attributes which are out of range within a Group

1 Baseline. This would quickly allow a system administrator to dismiss WEB-PATCHES as a  
2 problem area and allow him or her to look for other areas in which to find root cause of change.<sup>4</sup>

3 Figure 4 illustrates the means with which to progress through the Baselines to identify the  
4 properties, or patterns, of the most intense change in the infrastructure. The group selected  
5 remains as it was in Fig 3, i.e., Web-GRP1. Since, as described in Fig 3, the User learned that  
6 the Baseline WEB-PATCHES had no changes, they move to another Baseline in an effort to  
7 identify a pattern of the change. Selection of Baseline (Fig 4 – 1.0), select the Baseline through  
8 which to filter the node group (this example provides a visualization of nodes in WEB-GRP1 as  
9 filtered through the attribute-value associations of NT-PERF). Node View (Fig 4 – 2.0) presents  
10 the group nodes with the status relative to the Baseline; Node View Pie Chart (Fig 4 – 3.0)  
11 continually provides visualization of the quantitative percentage of change in full population  
12 Group Selection Pie Chart (Fig 4 – 4.0) provides visualization of the quantitative percentage of  
13 change within the baseline for the selected set of nodes. Comparing the Node View Pie Chart to  
14 the Group View Pie Chart indicates quickly that the percentage of change is greater in the NT  
15 PERF Baseline than the greater population and indicates an area for further investigation.<sup>5</sup>

16 Figure 5 depicts the drill down from Figure 4, focusing specifically on the Node Group  
17 and Baseline selected at the point the User Drills Down. Node Group View (Fig 5 – 1.0),  
18 presents the selected group nodes, delineated by location, with the status relative to the Baseline.  
19 The drill-down view reduces the number of nodes in the map, while leaving the remainder of the  
20 screen and its corresponding functionality intact.

---

<sup>4</sup> Multiple Groups may be selected.

<sup>5</sup> Multiple Baselines may be selected.

1       Figure 6 illustrates alternate 3D views of Drill Down. 3D- Z Axis (Fig 6 – 1.0) is the  
2 power axis and can be configured by the User to represent any key aspect of the nodes being  
3 monitored (e.g. CPU Power (3of CPUs \* CPU Speed), # of Users, Revenue,)

4

5       Color

6       The color assigned to a node is determined using a weighted moving average. Increasing  
7 the time of the sampled data for each attribute creates an average. The greater the percentage of  
8 change against that average, the greater the deviation and the greater the color shift (e.g. Green to  
9 Red).

10       The delta time is used to compute a moving average for each sample. Time is actually the  
11 number of samples back in time, e.g., if the Daily sample is selected (as shown in Figure 6), a  
12 delta time of 5 equates to the average of the last five days. The maximum and minimum of the  
13 averages are used to compute the entire range of possibility.

14       For example, if a CPU attribute is selected and it is currently 25%, and the last five days  
15 it was: 90%, 10%, 50% 50% and 50%, the min is 10%, the max is 90% and the moving average  
16 is  $(90+10+30+35+50)/5 = 43\%$ . Since 25 is less than 43% it will be on the green scale where 10  
17 is bright green and 43 is the midway point to red. To compute the exact color of green on the  
18 scale,  $43-10$  is 33 and  $25-10 = 15$ , so  $15/33$  is the percentage of green on the scale. Figure 7  
19 depicts a graphical illustration of this point.

20       Figure 8 identifies the radio button selections for time comparison (Fig7 – 1.0) Daily,  
21 Weekly and Monthly. The timeframe can be customized by using the Custom Timeframe Button

1 (Fig 7 – 2.0), this customization will allow complex time selections like each Monday between 2  
2 PM and 5 PM. Sliding Sample Mean Time (Fig 7 – 3.0) is used to allow the end user to change  
3 the default moving average in the computation of changes for Metrics types of attributes.

4

5 **User Color Selection**

6 As shown in Figure 9, a user can change the colors in their view according to the user  
7 preferences.

8 Finally, Figure 10 illustrates an exemplary network/compute infrastructure having  
9 Managers (Fig 10 - 1.0, 2.0, 2.1, 2.2), Managers with Gateways (Fig 10 - 3.0), Gateways (Fig 1 -  
10 4.0), Managed Nodes with Agents (Fig 10 - 5.1, 5.2, 5.3 etc), Managed Nodes that are Agentless  
11 (Fig 10 - 6.0, 6.1, 6.2 etc), Software including application software, that can be managed like a  
12 node (Fig 10 - 7.0, 7.1 etc.), and Special Devices that can be managed (Fig 10 - 8.0, 8.1, etc).

13

14 **CONCLUSION**

15 Having now described embodiments of the present invention, it should be apparent to  
16 those skilled in the art that the foregoing is illustrative only and not limiting, having been  
17 presented by way of example only. All the features disclosed in this specification (including any  
18 accompanying claims, abstract, and drawings) may be replaced by alternative features serving the  
19 same purpose, and equivalents or similar purpose, unless expressly stated otherwise. Therefore,  
20 numerous other embodiments of the modifications thereof are contemplated as falling within the  
21 scope of the present invention as defined by the appended claims and equivalents thereto.

1        The techniques may be implemented in hardware or software, or a combination of the  
2        two. Specifically, the techniques may be implemented in computer programs executing on  
3        programmable computers that each include a processor, a storage medium readable by the  
4        processor (including volatile and non-volatile memory and/or storage elements), at least one  
5        input device and one or more output devices. Program code is applied to data entered using the  
6        input device to perform the functions described and to generate output information. The output  
7        information is applied to one or more output devices. Each program is preferably implemented  
8        in a high level procedural or object oriented programming language to communicate with a  
9        computer system, however, the programs can be implemented in assembly or machine language,  
10       if desired. In any case, the language may be a compiled or interpreted language. Each such  
11       computer program is preferably stored on a storage medium or device (e.g., CD-ROM, hard disk  
12       or magnetic diskette) that is readable by a general or special purpose programmable computer for  
13       configuring and operating the computer when the storage medium or device is read by the  
14       computer to perform the procedures described in this document. The invention may also be  
15       considered to be implemented as a computer-readable storage medium, configured with a  
16       computer program, where the storage medium so configured causes a computer to operate in a  
17       specific and predefined manner.

18

19